

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

Claims 1. - 3. (canceled)

Claims 4. (currently amended) ~~A method according to claim 3,~~ A method for allocating space among a plurality of queues in a buffer, comprising:  
sorting the plurality of queues of the buffer according to size, thereby to establish a sorted order of the queues;  
selecting at least one group of the queues comprising a given number of the queues in accordance with the sorted order;  
allocating a portion of the space in the buffer to the at least one group, responsive to the number of the queues in the at least one group; and  
accepting a data packet into one of the queues in the at least one group responsive to whether the data packet will cause the space occupied in the buffer by the queues in the group to exceed the allocated portion of the space;  
wherein selecting the at least one group comprises selecting for inclusion in the at least one group the queues that are largest among the plurality of queues;  
wherein allocating the portion of the space comprises setting a size of the space proportional to a sum of a harmonic series;  
wherein setting the size of the space comprises establishing a total maximum buffer size  $B_k$  for the  $k$  largest output queues, wherein  $B_k$  is substantially given by

$$B_k = \frac{M}{(\ln N + 1)} \sum_{i=1}^k \frac{1}{i}$$

wherein  $M$  is the total buffer space available to all queues,  $N$  is the number of output queues, and  $i$  is an integer index.

1 Claim 5. (original) A method according to claim 4, wherein selecting the number  
2 k of the queues comprises selecting a fixed number of the queues.

1 Claim 6. (original) A method according to claim 4, wherein k has a variable  
2 value.

1 Claim 7. (previously presented) A method according to claim 6, wherein k and  
2  $B_k$  are set in accordance with the equation:

$$3 \quad \underline{B_k} = \left(n + \frac{M}{c^{n+1}}\right) \cdot \frac{M}{\log c N}$$

4 wherein n, c and m are parameters such that  $1 < c < N$ , and n is the largest integer  
5 such that m and k satisfy the conditions

$$6 \quad 0 \leq m \leq cn+1$$

7 and

$$8 \quad k = \sum_{i=1}^n c_i + m.$$

1 Claim 8. (canceled)

1 Claim 9. (currently amended) ~~A method according to claim 8,~~ A method for  
2 allocating space among a plurality of queues in a buffer, comprising:  
3 sorting the plurality of queues of the buffer according to size, thereby to establish  
4 a sorted order of the queues;  
5 selecting at least one group of the queues comprising a given number of the  
6 queues in accordance with the sorted order;  
7 allocating a portion of the space in the buffer to the at least one group, responsive  
8 to the number of the queues in the at least one group; and

9                    accepting a data packet into one of the queues in the at least one group responsive  
10 to whether the data packet will cause the space occupied in the buffer by the queues in the group  
11 to exceed the allocated portion of the space;

12                    wherein buffer space is allocated within said the at least one group of queues in  
13 accordance with a set of conditions which functionally define a known buffer management  
14 policy;

15                    wherein ~~said~~ the known buffer management policy is chosen from a group  
16 consisting of complete partitioning (CP), sharing with maximum queue lengths (SMXQ), sharing  
17 with minimum allocation (SMA), sharing with maximum queue length and minimum allocation  
18 (SMQMA), and dynamic threshold (DT).

1                    Claims 10.-13. (canceled)

1                    Claim 14. (currently amended) ~~A method according to claim 13, A method for~~  
2 allocating space among N output queues in a buffer of size M, which comprises:

3                    sorting the plurality of queues of the buffer according to size, thereby to establish  
4 a sorted order of the queues;

5                    selecting a number k of the N output queues in accordance with the sorted order;  
6                    establishing a total maximum buffer space of  $B_k$  for the number k of the N output  
7 queues, wherein  $B_k < M$ ;

8                    ascertaining whether acceptance of an arriving packet destined for one of the  
9 number k of the N output queues will cause the space in the buffer used by the k queues together  
10 to exceed  $B_k$ ; and

11                    if the acceptance of an arriving packet destined for one of the k queues will cause  
12 the space used by the k queues together to exceed  $B_k$ , rejecting the packet;

13                    wherein the selecting a number k of the N output queues comprises selecting a  
14 number k of the N output queues that are largest among the N output queues;

15                    wherein establishing the total maximum buffer space comprises setting the  
16 maximum buffer size such that B is substantially given by

17 
$$B_k = \frac{M}{(\ln N + 1)} \sum_{i=1}^k \frac{1}{i}$$

18 wherein M is the total buffer space available to all queues, and N is the number of  
19 output queues, and i is an integer index which refers to an ordinal ranking of the queue size.

1 Claim 15. (canceled)

1 Claim 16. (currently amended) ~~A method according to claim 15,~~ A method for  
2 allocating space among N output queues in a buffer of size M, which comprises:  
3 sorting the plurality of queues of the buffer according to size, thereby to establish  
4 a sorted order of the queues;  
5 selecting a number k of the N output queues in accordance with the sorted order;  
6 establishing a total maximum buffer space of  $B_k$  for the number k of the N output  
7 queues, wherein  $B_k < M$ ;  
8 ascertaining whether acceptance of an arriving packet destined for one of the  
9 number k of the N output queues will cause the space in the buffer used by the k queues together  
10 to exceed  $B_k$ ; and  
11 if the acceptance of an arriving packet destined for one of the k queues will cause  
12 the space used by the k queues together to exceed  $B_k$ , rejecting the packet;  
13 wherein the selecting a number k of the N output queues comprises selecting a  
14 number k of the N output queues that are largest among the N output queues;  
15 wherein the buffer space is allocated within the number k of the N output queues  
16 in accordance with a set of conditions which functionally define a known buffer management  
17 policy;  
18 wherein said the known buffer management policy is chosen from a group  
19 consisting of complete partitioning (CP), sharing with maximum queue lengths (SMXQ), sharing  
20 with minimum allocation (SMA), sharing with maximum queue length and minimum allocation  
21 (SMQMA), and dynamic threshold (DT).

Claims 17.- 18. (canceled)

Claim 19. (currently amended) ~~A method according to claim 18,~~ A method for allocating space among N output queues in a buffer of size M, which comprises:  
sorting the plurality of queues of the buffer according to size, thereby to establish a sorted order of the queues;  
selecting a number k of the N output queues in accordance with the sorted order;  
establishing a total maximum buffer space of  $B_k$  for the number k of the N output queues, wherein  $B_k < M$ ;  
ascertaining whether acceptance of an arriving packet destined for one of the number k of the N output queues will cause the space in the buffer used by the k queues together to exceed  $B_k$ ; and  
if the acceptance of an arriving packet destined for one of the k queues will cause the space used by the k queues together to exceed  $B_k$ , rejecting the packet;  
wherein the value of k is variable;  
wherein the values of k and  $B_k$  are set in accordance with the equation

$$B_k = \left(n + \frac{M}{c^{n+1}}\right) \cdot \frac{M}{\log c N}$$

wherein n, c and m are parameters such that  $1 < c < N$ , and n is the largest integer such that in and k satisfy the conditions:

$$0 \leq m \leq cn+1$$

and

$$k = \sum_{i=1}^n ci + m.$$

Claims 20. - 21. (canceled)

Claim 22. (currently amended) ~~A switch according to claim 21,~~ A shared memory switch comprising:

3                   a memory configured to provide a buffer space that is adapted to contain a  
4 plurality of output queues; and  
5                   a controller, configured to:  
6                   sort all the queues of the buffer according to size, thereby to establish a sorted  
7 order of the queues,  
8                   allocate the space in the buffer to the output queues in accordance with the sorted  
9 order such that a portion of the space is allocated to a group of the output queues comprising a  
10 given number of the output queues responsive to the given number of the output queues in the  
11 group, and  
12                   accept a data packet into one of the output queues in the group responsive to  
13 whether the data packet will cause the space occupied in the buffer by the output queues in the  
14 group to exceed the allocated portion of the space;  
15                   wherein the portion of the space is allocated by setting a size of the space  
16 proportional to a sum of a harmonic series;  
17                   wherein the portion of the space is allocated for the k largest output queues and  
18 the total maximum buffer size  $B_k$  for the k largest output queues is substantially given by  
19 
$$B_k = \frac{M}{(1n \quad N+1)} \sum_{i=1}^k \frac{1}{i}$$
  
20                   wherein i is an integer index which refers to an ordinal ranking of the queue  
21 size.

1                   Claim 23. (currently amended) A switch according to claim 22, wherein ~~said~~ the  
2 portion of the space is allocated for the k largest output queues and the total maximum buffer  
3 size  $B_k$  for ~~said~~ the k largest output queues is substantially given by the equation

$$B_k = \left(n + \frac{M}{c^{n+1}}\right) \cdot \frac{M}{\log c N}$$

5                   wherein n, c and m are parameters such that  $1 < c < N$ , and n is the largest integer  
6 such that m and k satisfy the conditions

$$0 \leq m \leq cn+1$$

and

$$k = \sum_{i=1}^n c_i + m.$$

Claim 24. (canceled)

Claim 25. (currently amended) ~~A switch according to claim 24, A shared~~  
memory switch comprising:  
a memory configured to provide a buffer space that is adapted to contain a  
plurality of output queues; and  
a controller, configured to:  
sort all the queues of the buffer according to size, thereby to establish a sorted  
order of the queues,  
allocate the space in the buffer to the output queues in accordance with the sorted  
order such that a portion of the space is allocated to a group of the output queues comprising a  
given number of the output queues responsive to the given number of the output queues in the  
group, and  
accept a data packet into one of the output queues in the group responsive to  
whether the data packet will cause the space occupied in the buffer by the output queues in the  
group to exceed the allocated portion of the space;  
wherein the switch has N output lines and N corresponding output queues,  
wherein the given number is denoted by k and the portion of the space available for the k largest  
output queues is denoted by  $B_k$ , wherein  $B_k < M$ ;  
wherein  $B_k$  is substantially given by the equation

$$B_k = \frac{M}{(1n \quad N+1)} \sum_{i=1}^k \frac{1}{i}$$

wherein i is an integer index which refers to the ordinal ranking of the queue in  
terms of the queue size.

Claims 26. - 28. (canceled)

Claim 29. (currently amended) ~~The device according to claim 28,~~ A device for allocating space among a plurality of queues in a buffer, comprising:  
logic configured to sort the plurality of queues of the buffer according to size,  
thereby to establish a sorted order of the queues;  
logic configured to select at least one group of the queues comprising a given number of the queues in accordance with the sorted order;  
logic configured to allocate a portion of the space in the buffer to the at least one group, responsive to the number of the queues in the at least one group; and  
logic configured to accept a data packet into one of the queues in the at least one group responsive to whether the data packet will cause the space occupied in the buffer by the queues in the group to exceed the allocated portion of the space;  
wherein the logic configured to select the at least one group comprises logic configured to select for inclusion in the at least one group the queues that are largest among the plurality of queues;  
wherein the logic configured to allocate the portion of the space comprises setting a size of the space proportional to a sum of a harmonic series:  
wherein the logic configured to set the size of the space comprises logic configured to establish a total maximum buffer size  $B_k$  for the  $k$  largest output queues, wherein  $B_k$  is substantially given by

$$B_k = \frac{M}{(\ln N + 1)} \sum_{i=1}^k \frac{1}{i}$$

wherein  $M$  is the total buffer space available to all queues,  $N$  is the number of output queues, and  $i$  is an integer index.

Claim 30. (previously presented) The device according to claim 29, wherein selecting the number  $k$  of the queues comprises selecting a fixed number of the queues.



1                    Claim 32. (previously presented) The device according to claim 31, wherein k  
2    and B<sub>k</sub> are set in accordance with the equation:

4                    wherein  $n$ ,  $c$  and  $m$  are parameters such that  $1 < c < N$ , and  $n$  is the largest integer  
5                    such that  $m$  and  $k$  satisfy the conditions

7 and

1                      Claim 33. (canceled)

1                   Claim 34. (currently amended) ~~The device according to claim 33,~~ A device for  
2 allocating space among a plurality of queues in a buffer, comprising:  
3                   logic configured to sort the plurality of queues of the buffer according to size,  
4 thereby to establish a sorted order of the queues;  
5                   logic configured to select at least one group of the queues comprising a given  
6 number of the queues in accordance with the sorted order;  
7                   logic configured to allocate a portion of the space in the buffer to the at least one  
8 group, responsive to the number of the queues in the at least one group; and  
9                   logic configured to accept a data packet into one of the queues in the at least one  
10 group responsive to whether the data packet will cause the space occupied in the buffer by the  
11 queues in the group to exceed the allocated portion of the space;  
12                   wherein the logic configured to select the at least one group comprises logic  
13 configured to select for inclusion in the at least one group the queues that are largest among the  
14 plurality of queues;  
15                   wherein buffer space is allocated within the at least one group of queues in  
16 accordance with a set of conditions which functionally define a known buffer management  
17 policy;  
18                   wherein ~~said~~ the known buffer management policy is chosen from a group  
19 consisting of complete partitioning (CP), sharing with maximum queue lengths (SMXQ), sharing  
20 with minimum allocation (SMA), sharing with maximum queue length and minimum allocation  
21 (SMQMA), and dynamic threshold (DT).

1                   Claims 35.- 39. (canceled)

1                   Claim 40. (currently amended) ~~A method according to claim 39,~~ A method for  
2 allocating space among a plurality of queues in a buffer, comprising:  
3                   selecting at least one group of the queues comprising a number of queues in the  
4 plurality of queues based on a sorting of the plurality of queues;

5                    allocating a portion of the space in the buffer to the at least one group based on  
6                    the number of the queues in the at least one group; and  
7                    accepting a data packet into one of the queues in the at least one group responsive  
8                    to whether the data packet will cause the space occupied in the buffer by the queues in the group  
9                    to exceed the allocated portion of the space;  
10                   wherein selecting the at least one group comprises selecting for inclusion in the at  
11                   least one group the queues that are largest among the plurality of queues;  
12                   wherein allocating the portion of the space comprises setting a size of the space  
13                   proportional to a sum of a harmonic series;  
14                   wherein setting the size of the space comprises establishing a total maximum  
15                   buffer size  $B_k$  for the  $k$  largest output queues, wherein  $B_k$  is substantially given by

$$B_k = \frac{M}{(\ln N + 1)} \sum_{i=1}^k \frac{1}{i}$$

17                   wherein  $M$  is the total buffer space available to all queues,  $N$  is the number of  
18                   output queues, and  $i$  is an integer index.

1                   Claim 41. (previously presented) A method according to claim 40, wherein  
2                   selecting the number  $k$  of the queues comprises selecting a fixed number of the queues.

1                   Claim 42. (previously presented) A method according to claim 40, wherein  $k$  has  
2                   a variable value.

1                   Claim 43. (previously presented) A method according to claim 42, wherein  $k$  and  
2                    $B_k$  are set in accordance with the equation:

$$B_k = \left(n + \frac{M}{c^{n+1}}\right) \cdot \frac{M}{\log c N}$$

4                   wherein  $n$ ,  $c$  and  $m$  are parameters such that  $1 < c < N$ , and  $n$  is the largest integer  
5                   such that  $m$  and  $k$  satisfy the conditions

$$0 \leq m \leq cn + 1$$

and

$$k = \sum_{i=1}^n c_i + m.$$

Claim 44. (canceled)

Claim 45. (currently amended) ~~A method according to claim 44,~~ A method for allocating space among a plurality of queues in a buffer, comprising:  
selecting at least one group of the queues comprising a number of queues in the plurality of queues based on a sorting of the plurality of queues;  
allocating a portion of the space in the buffer to the at least one group based on the number of the queues in the at least one group; and  
accepting a data packet into one of the queues in the at least one group responsive to whether the data packet will cause the space occupied in the buffer by the queues in the group to exceed the allocated portion of the space;  
wherein selecting the at least one group comprises selecting for inclusion in the at least one group the queues that are largest among the plurality of queues;  
wherein ~~said~~ the known buffer management policy is chosen from a group consisting of complete partitioning (CP), sharing with maximum queue lengths (SMXQ), sharing with minimum allocation (SMA), sharing with maximum queue length and minimum allocation (SMQMA), and dynamic threshold (DT).

Claims 46. - 48. (canceled)

Claim 49. (previously presented) The method of claim 6, wherein the variable value of k is determined based on a parametric harmonic policy.

Claim 50. (previously presented) The method of claim 14, wherein setting the size of the space comprises establishing a total buffer size based on a number, N, of the plurality

3 of queues; a number, k, of the queues in the at least one group; and a ranked order of the queues  
4 in the at least one group based on queue length.

1 Claims 51.- 55. (canceled)

1 Claim 56. (previously presented) The device of claim 31, wherein the variable  
2 value of k is determined based on a parametric harmonic policy.

1 Claim 57. (canceled)

2 Claim 58. (previously presented) The method of claim 42, wherein the variable  
3 value of k is determined based on a parametric harmonic policy.